

Listing of the Claims:

1. (Currently Amended) A method of storing streamed presentation data within a container file, the method executing on a consumer digital content playback device, the method comprising:

receiving one or more data streams from each of one or more presentation sources within the presentation;

creating within the container file, a virtual file for each of the one or more presentation sources;

temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the presentation source;

determining a container file size of the container file;

temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size; and

rendering at least one of said one or more data streams.

2. (Original) The method of claim 1, wherein the additional data from the first data stream is stored in place of at least a portion of the first data if the container file size is equal to or exceeds the identified maximum buffer size.

3. (Original) The method of claim 1, further comprising: temporarily storing second data associated with a second data stream of the first presentation source in association with the first virtual file; and temporarily storing additional data from the second data stream in place of at least a portion of the second data stored in association with the first virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

4. (Original) The method of claim 3, further comprising: rendering one of the first and second data streams in real-time contemporaneous with the storing of at least one of the first and second data streams.

5. (Original) The method of claim 3, further comprising: temporarily storing data associated with a third data stream of a second presentation source in association with a second virtual file; and temporarily storing additional data from the third data stream in place of at least a portion of the data stored in association with the second virtual file if the container file size is within the predetermined range of the identified maximum buffer size.
6. (Original) The method of claim 1, wherein the maximum buffer size is proportional to an amount of time indicated via a user interface.
7. (Original) The method of claim 1, wherein the maximum buffer size is dynamically increased during the storing of data from the first data stream.
8. (Original) The method of claim 1, wherein the first data and additional data are stored in a native packet format prior to a decoding process.
9. (Original) The method of claim 1, wherein each virtual file comprises: at least a first data block; and a file descriptor block containing at least a seek index and a seek index granularity, wherein the seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks.
10. (Original) The method of claim 9, wherein the additional data is stored in place of the first data beginning with the first data block and continuing with successive data blocks of the first virtual file.
11. (Original) The method of claim 9, wherein if the container file size is within the predetermined range of the identified maximum buffer size, the seek index granularity is increased so as to increase data block size without changing the number of seek index entries.
12. (Original) The method of claim 9, further comprising: receiving a user indication identifying a location corresponding to a time (T) within the presentation; identifying a seek position for each virtual file, wherein each seek position is determined by dividing time (T)

by the seek granularity for the corresponding virtual file; and contemporaneously rendering in real-time, data stored in each virtual file at the respective seek positions.

13. (Currently Amended) A machine readable medium having machine executable instructions, which when executed on a consumer digital content playback device operate to implement a method comprising:

receiving one or more data streams from each of one or more presentation sources within a presentation;

creating within a container file, a virtual file for each of the one or more presentation sources;

temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the presentation source;

determining a container file size of the container file;

temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size; and

rendering at least one of said one or more data streams.

14. (Original) The machine readable medium of claim 13, wherein the additional data from the first data stream is stored in place of at least a portion of the first data if the container file size is equal to or exceeds the identified maximum buffer size.

15. (Original) The machine readable medium of claim 13, further comprising instructions to temporarily store second data associated with a second data stream of the first presentation source in association with the first virtual file; and temporarily store additional data from the second data stream in place of at least a portion of the second data stored in association with the first virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

16. (Original) The machine readable medium of claim 15, further comprising instructions to render one of the first and second data streams in real-time contemporaneous with the storing of at least one of the first and second data streams.
17. (Original) The machine readable medium of claim 15, further comprising instructions to: temporarily store data associated with a third data stream of a second presentation source in association with a second virtual file; and temporarily store additional data from the third data stream in place of at least a portion of the data stored in association with the second virtual file if the container file size is within the predetermined range of the identified maximum buffer size.
18. (Original) The machine readable medium of claim 13, wherein the maximum buffer size is proportional to an amount of time indicated via a user interface.
19. (Original) The machine readable medium of claim 13, wherein the maximum buffer size is dynamically increased during the storing of data from the first data stream.
20. (Original) The machine readable medium of claim 13, wherein the first data and additional data are stored in a native packet format prior to a decoding process.
21. (Original) The machine readable medium of claim 13, wherein each virtual file comprises: at least a first data block; and a file descriptor block containing at least a seek index and a seek index granularity, wherein the seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks.
22. (Original) The machine readable medium of claim 21, wherein the additional data is stored in place of the first data beginning with the first data block and continuing with successive data blocks of the first virtual file.
23. (Original) The machine readable medium of claim 21, wherein if the container file size is within the predetermined range of the identified maximum buffer size, the seek index

granularity is increased so as to increase data block size without changing the number of seek index entries.

24. (Original) The machine readable medium of claim 21, further comprising instructions to receive a user indication identifying a location corresponding to a time (T) within the presentation; identify a seek position for each virtual file, wherein each seek position is determined by dividing time (T) by the seek granularity for the corresponding virtual file; and contemporaneously render in real-time, data stored in each virtual file at the respective seek positions.